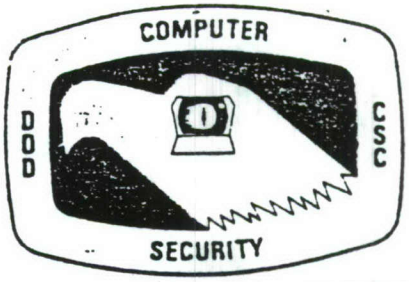
THE ACCESS CONTROL FACILITY 2 (ACF2) RELEASE 3.1.3. (FINAL EVALUATION REPORT)

(U.S.) Department of Defense
Ft. Mead, MD

Aug 84

CSC-EPL-84/002



FINAL EVALUATION REPORT

The Access Control Facility 2 (ACF2)

Release 3.1.3

3 August 1984

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public release; Distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CSC-EPL-84/002	5. MONITORING ORGANIZATION REPORT NUMBER(S) S-226,529
6a. NAME OF PERFORMING ORGANIZATION Department of Defense Computer Security Center 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) 9800 Savage Road Ft. Meade, MD 20755-6000	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT dentification number
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) SKK Access Control Facility 2 (ACF2) Release 3.1.3 Final Evaluation Report	
12. PERSONAL AUTHOR(S) ISRAEL, Howard; LAFOUNTAIN, Steven	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) 84/08/03 15. PAGE COUNT
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Trusted Computer System Evaluation Criteria The Access Control Facility 2 C2 EPL DoDCSC ACF2 SKK MVS

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The security features of ACF2/MVS were evaluated against the requirements specified by the DoD Trusted Computer System Evaluation Criteria dated 15 August 1983 and found to satisfy all the requirements of evaluation class C2.

The Department of Defense Computer Security Center (DoDCSC) was established in January 1981 to encourage the widespread availability of trusted computer systems for use by facilities processing classified or other sensitive data.

In the first quarter of FY83, SKK Inc. requested that the DoDCSC evaluate their commercially available product, the Access Control Facility 2 (ACF2) Release 3.1.3 for IBM's OS/VS2 MVS operating system. MVS is an IBM operating system for IBM's 303x, 308x, 4341, 370/158, and 370/168 processors.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☒ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Mario Tinto	22b. TELEPHONE NUMBER (Include Area Code) 301-859-6044	22c. OFFICE SYMBOL C12

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

CSC-EPL-84/002
Library No. S-226,529

FOREWORD

This publication, SKK Access Control Facility 2 (ACF2) Release 3.1.3 Final Evaluation Report, is being issued by the DoD Computer Security Center under the authority and in accordance with DoD Directive 5215.1, "Computer Security Evaluation Center." The purpose of this report is to document the results of the formal evaluation of SKK's ACF2 add-on security package. The requirements stated in this report are taken from the Department of Defense Trusted Computer System Evaluation Criteria dated 15 August 1983.

Approved:

Robert L. Brotzman
Director
DoD Computer Security Center

3 August 1984

Table of Contents

EVALUATION TEAM MEMBERS iii

EXECUTIVE SUMMARY iv

1 INTRODUCTION 1

1.1 BACKGROUND 1

1.2 EVALUATION PROCESS 1

1.3 EVALUATION ENVIRONMENT 2

1.4 DOCUMENT ORGANIZATION. 4

2 ACF2 vs. THE CRITERIA AT CLASS C2. 5

2.1 DISCRETIONARY ACCESS CONTROL 5

2.2 OBJECT REUSE 6

2.3 IDENTIFICATION AND AUTHENTICATION. 6

2.4 AUDIT. 8

2.5 SYSTEM ARCHITECTURE. 9

2.6 SYSTEM INTEGRITY 12

2.7 SECURITY TESTING 14

2.8 DOCUMENTATION. 14

3 DEFICIENCIES AGAINST CLASS B1 REQUIREMENTS 17

4 THOSE REQUIREMENTS WHERE ACF2 EXCEEDS CLASS C2 18

4.1 DISCRETIONARY ACCESS CONTROL 18

5 EVALUATORS' COMMENTS 19

6 CONCLUSIONS. 20

REFERENCES 21

GLOSSARY 22

EVALUATION SUMMARY CHART 23

APPENDIX A (TEST PLAN) a-1

APPENDIX B (MVS System Utilities). b-1

EVALUATION TEAM MEMBERS

Howard M. Israel
Steven M. La Fountain
William M. Lake
Jonathan R. Monsein
Suzanne S. O'Connor
Lisa A. Wurm
DoD Computer Security Center
9800 Savage Road
Fort George G. Meade, MD 20755-6000

David Bohrer
Jill Seeger
902nd Military Intelligence Group
Fort George G. Meade, MD 20755-6000

EXECUTIVE SUMMARY

The security protection provided by SKK's ACF2 add-on package release 3.1.3 running with IBM's Multiple Virtual Storage/System Product (MVS/SP) 1.3.3 operating system has been evaluated by the Department of Defense Computer Security Center (DoDCSC). ACF2, as evaluated, included the code described in SKK's NOTE #7, SUBJECT: AUTOMATIC ERASE. This code is available from SKK. The security features of ACF2/MVS were evaluated against the requirements specified by the Department of Defense Trusted Computer System Evaluation Criteria (the Criteria) dated 15 August 1983.

The DoDCSC evaluation team has determined that the highest class at which ACF2/MVS satisfies all the requirements of the Criteria is class C2 and therefore ACF2/MVS has been assigned a class C2 rating. It should be noted, however, that this rating is contingent upon the utilization of code described in SKK's NOTE #7.

A system that has been rated as being a C division system contains the features and assurances described in the Criteria. There is no assurance that a C division system is free of flaws that would allow the subversion or bypassing of the advertised security mechanisms through penetration methods.

The evaluation of ACF2 was performed only with IBM's MVS operating system. The integrity of ACF2 is dependent upon the integrity of the MVS system itself.

SECTION 1.0

INTRODUCTION

1.1 BACKGROUND

The Department of Defense Computer Security Center (DoDCSC) was established in January 1981 to encourage the widespread availability of trusted computer systems for use by facilities processing classified or other sensitive data.

In the first quarter of FY 83, SKK, Inc. requested that the DoDCSC evaluate their commercially available add-on security package, ACF2 Release 3.1.3, for the OS/VS2 MVS and VS1 operating systems. MVS and VS1 are IBM operating systems for its 303x, 308x, 4341, 370/158 and 370/168 processors.

1.2 EVALUATION PROCESS

The DoDCSC established an evaluation team and the evaluation commenced with a meeting on 13 December 1982 at the DoDCSC at Ft. Meade, MD. At this meeting SKK and the DoDCSC committed themselves to complete a formal evaluation of ACF2. The DoDCSC evaluation team would examine and test the security features of ACF2 against the final draft of the Trusted Computer System Evaluation Criteria that was to be issued in January of 1983 (the evaluation was later updated to use the 15 August version of the Criteria). Following the completion of the examination and testing, the evaluation team would issue a report that detailed the evaluation. ACF2 would then be assigned a rating and placed on the DoDCSC's Evaluated Products List (EPL).

The evaluation team commenced their examination of ACF2 by attending ACF2 and MVS classes, reviewing ACF2 documentation, and gaining hands-on experience with ACF2.

A preliminary working paper was produced which mapped the evaluation team's understanding of the features of ACF2 into the requirements of the evaluation Criteria and presented an initial summary of the evaluation testing that was planned for the third quarter of FY83. This working paper was sent to SKK for their review and comment.

The evaluation team met with SKK on 15 February 1983 at SKK offices in Rosemont, IL to receive SKK's comments on the preliminary working paper.

A detailed test plan was developed and sent to SKK for their review. The evaluation team then met with SKK in May to review documentation and procedures of SKK's own internal testing of ACF2 Release 3.1.3.

The security features functional testing was conducted at Harry Diamond Laboratories (HDL) on a system 370/168 running MVS/SP 1.3.3 with 4 Megabytes of real storage, Job Entry Subsystem2 (JES2) Release 4.1 (maintenance level 8111), ACF/TCAM Release 2.3 (maintenance level 8303), ACF2 Release 3.1.3, Time Sharing Option (TSO), and the System Productivity Facility (SPF). This testing was conducted between 4-18 August 1983. At the completion of this test session, it was determined that ACF2 satisfied all of the class C2 requirements except for object reuse.

In mid-January of 1984, a draft of the final evaluation report was sent to SKK detailing the results of the evaluation. At this time, SKK asked that the evaluation be halted until they could produce a modification intended to correct this problem. This modification was received by the evaluation team in early March 1984.

A second test session was then held at HDL during July 1984 to test this modification. After this session of testing, it was determined that ACF2 now satisfied all of the requirements of class C2, with no exceptions. After completing the functional testing, the evaluation team prepared this final report.

1.3 **EVALUATION ENVIRONMENT**

ACF2 is an add-on security package for the MVS and VS1 operating systems that provides, by default, controlled access to system resources. ACF2 intercepts cause the operating system to call ACF2 for authorization checking at every LOGON, READ, WRITE, ALLOCATE, SCRATCH, RENAME and catalog processing. Numerous options, selectable at installation time, provide a high degree of discretionary security. The owner of a resource is allowed to specify and control access to that resource. This access can be granted in various degrees (READ, WRITE, ALLOCATE, and EXECUTE).

READ - allows for the reading of the specified data sets

WRITE - allows for the writing to the specified data sets

ALLOCATE - for data sets, allows for the scratch, rename or catalog

- for volumes, allows for the allocation of new data sets

EXECUTE - allows for the execution of the specified program (DOES NOT ALLOW READ).

These access modes can be granted or denied by specifying the ALLOW, PREVENT, or LOG parameter in the appropriate access rule. The LOG parameter allows the specified access but records an entry in the SMF data sets indicating that the access was granted.

ACF2 also protects access to the system itself. ACF2 requires that each user of the system be defined to ACF2 through a unique Logonid. Each user must supply his unique Logonid and a password before being allowed access to the system. The installation has the ability to define syntax rules with which all passwords must comply. These rules govern the length and the alphanumeric syntax of the password. The installation may also define maximum and minimum time intervals within which a user must change his password and also the maximum number of consecutive times that a user can supply an invalid password before his Logonid is suspended. When a Logonid is suspended by ACF2, the operator has the option to allow one additional attempt, if that attempt also fails, the Logonid must be reset by a user with the SECURITY attribute (a security administrator) before that Logonid is again allowed access to the system.

ACF2 provides protection by default. All data sets created under ACF2 are not accessible by any user until the owner writes an access rule allowing access.

ACF2 provides extensive audit capabilities by writing to the MVS System Management Facility (SMF) data sets. The default is to write a record to SMF whenever a detected, unauthorized access attempt is made. There is also the ability to log authorized accesses. This is done by specifying the appropriate entry in the access rules.

The report writing utilities can be used to obtain reports on detected unauthorized access attempts, successful authorized accesses, and various other activities.

For the maximum security possible, users should be defined to ACF2 with the minimum amount of authority that will allow them to efficiently and effectively use the system. Passwords should be enforced and regular password change should be required. The site should install and use the code supplied by SKK that allows for the erasure of all storage areas before these areas are released to system users. Finally, the ACF2 audit utilities should be used to the maximum, reasonable extent and the report utilities should be run regularly to obtain information about which users are utilizing what resources and how.

1.4 DOCUMENT ORGANIZATION

This report consists of six major sections. Section 2 provides the class C2 requirements, as stated in the Criteria, and describes the functions of the security features that resulted in ACF2/MVS being assigned a class C2 rating. Section 3 details how ACF2 compares against the B1 requirements of the Criteria and explains why ACF2 could not be given a B1 rating. Section 4 describes ACF2/MVS features that exceed the requirements of class C2 (the class at which it is rated). Section 5 presents comments by the evaluation team concerning specific ACF2/MVS features and a recommended mode of implementing ACF2. Section 6 presents the conclusions of this report. Appendix A provides a description of the testing conducted for the evaluation. Appendix B provides a list of utilities provided with the MVS operating system.

SECTION 2

ACF2 vs. CLASS C2 REQUIREMENTS

THIS SECTION ADDRESSES THE CLASS FOR WHICH ACF2 SATISFIES ALL REQUIREMENTS OF THE CRITERIA.

2.1 DISCRETIONARY ACCESS CONTROL

Requirement:

The Trusted Computing Base (TCB) shall define and control access between named users and named objects (e.g., files and programs) in the ADP system. The enforcement mechanism (e.g., self/group/public controls, access control lists) shall allow users to specify and control sharing of those objects by named individuals or defined groups of individuals or by both. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of including or excluding access to the granularity of a single user. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users.

Satisfied By:

All resources (e.g., files, disk and tape volumes, subsystems, etc.) in the ACF2/MVS system are protected by default. ACF2 provides controlled access between named users and objects by means of access rules and generalized resource rules. All accesses to the system, to data, and to resources are prevented unless specified in an access rule. These rules are stored in the ACF2 rules data base.

User's may control the sharing of their own data between individual users or groups of users by entering a rule into the rules data base. The owner of a resource may specify who can access that resource, in what ways (access methods), from what source (terminals, remote readers) and when (day, date, time). Any time that a rule is not found to allow a requested access, an audit record is written to the System Management Facility (SMF) data sets and the requested access is denied. The owner of a resource may also specify auditing of the accesses to any resource that he owns by entering the LOG parameter in the appropriate access rule. (A security administrator can specify auditing for any resource under his control). The LOG parameter allows the specified access but records an entry in the SMF records indicating that the access was made.

Testing verified that resources are protected by default and access must be granted by the owner of the resource or a security administrator before any access attempt to that resource will be allowed. A security administrator can only write access rules for those resources under his control.

2.2 OBJECT REUSE

Requirement:

When a storage object is initially assigned, allocated, or reallocated to a subject from the Trusted Computing Base's pool of unused storage objects, the TCB shall assure that the object contains no data for which the subject is not authorized.

Satisfied By:

When pages of main memory are allocated to a user, they are automatically cleared by the MVS system before being released to the requesting user.

Storage areas on secondary storage devices are erased automatically by ACF2 if the installation is utilizing the code supplied by SKK to provide this feature (this feature is described in SKK NOTE #7, SUBJECT: AUTOMATIC ERASE). This is accomplished through the use of ACFERASE and ACFDEL, two ACF2 subroutines, which were always present in the ACF2 system but were only user invokable (i.e., they were not automatically invoked by the system at the time of data set deletion). The system administrator, through system provided exit points, now has the ability to provide for the erasure of data from all data set types present in the MVS system.

2.3 IDENTIFICATION AND AUTHENTICATION

Requirement:

The Trusted Computing Base (TCB) shall require users to identify themselves to it before beginning to perform any other actions that the TCB is expected to mediate. Furthermore, the TCB shall use a protected mechanism (e.g., passwords) to authenticate the user's identity. The TCB shall protect authentication data so that it cannot be accessed by any unauthorized user. The TCB shall be able to enforce individual accountability by providing the capability to uniquely identify each individual ADP system user. The TCB shall also provide the capability of associating this identity with all auditable actions taken by that individual.

Satisfied By:

System access is mediated by ACF2 at all TSO LOGONs, CICS and IMS Sign-ons, Batch Job Initiations, etc. ACF2 checks the Logonid (which must be unique) and authenticates that Logonid through the use of a password. ACF2 also checks to verify that the Logonid is allowed to use the input source and that the Logonid is allowed access to the system at this time.

Users are defined to ACF2 through unique Logonid Records. These Logonid Records are created via the INSERT subcommand of the ACF command. User attributes can be changed by the security administrator through the use of the CHANGE subcommand of the ACF command. Only a user with the ACCOUNT attribute can add a Logonid to the system or change the security relevant fields of a Logonid record.

ACF2 enforces password change and allows the installation to specify the minimum and maximum amounts of time that must pass before the password can/must be changed. The installation also has the option to specify an alphanumeric syntax (length and alphanumeric structure) with which all passwords must comply. Passwords are stored in encrypted form in the ACF2 Logonid data base. Passwords are never stored within the system in plain text form. When the plain text is input by the user, it is one-way encrypted and the encrypted form is compared to what is in the Logonid data base.

The installation also has the option to specify the number of times that a password can be supplied incorrectly before the password is expired by ACF2 (a user cannot access the system with an expired password). If a password has been expired by ACF2, it must be reset by the security administrator before the user may again gain access to the system. If the password has expired because the password change interval has passed, the user has the ability to, and is forced to change his password before he is again allowed access to the system. ACF2 also provides the installation with the option of retaining a password history for each user. The installation can then ensure that users do not change their password back to a previously used password before some specified time interval has passed. Messages are sent to the system and/or security console concerning password violations.

TSO passwords are not echoed back to the terminal screen. At no time during the testing were there any indications of TSO password exposure. However, as a caution, physical security should be enforced in order to prevent password exposure from batch Job Control Language (JCL) files. JCL card decks contain the password on a card, making physical card decks a potential exposure. Note that when a user is processing a batch job, or editing a batch stream file, passwords may be displayed on the

screen unless these passwords are suppressed. As a caution, if the RESTRICT attribute is used without other restrictions upon a user, the individual accountability requirement may be violated. Therefore, users possessing the RESTRICT attribute should be restricted by the use of other attributes (i.e., SUBAUTH and PROGRAM).

2.4 **AUDIT**

Requirement:

The TCB shall be able to create, maintain, and protect from modification or unauthorized access or destruction an audit trail of accesses to the objects it protects. The audit data shall be protected by the TCB so that read access to it is limited to those who are authorized for audit data. The TCB shall be able to record the following types of events: use of identification and authentication mechanisms, introduction of objects into a user's address space (e.g., file open, program initiation), deletion of objects, and actions taken by computer operators or system administrators and/or system security officers. For each recorded event, the audit record shall identify: date and time of the event, user, type of event, and success or failure of the event. For identification/authentication events the origin of request (e.g., terminal ID) shall be included in the audit record. For events that introduce an object into a user's address space and for object deletion events the audit record shall include the name of the object. The ADP system administrator shall be able to selectively audit the actions of any one or more users based on individual identity.

Satisfied By:

ACF2 creates, maintains, and protects a record of accesses to the system and to protected resources. This record identifies the user, resource, type of access attempted or obtained, the input source, and the time and date of access. Recording of accesses may be selectively invoked by the use of the LOG parameter in rules (which will keep track of accesses to a resource) and through the fields of the Logonid Record such as TRACE, MONITOR, and TSO-TRC (these will keep track of accesses by a particular user). Access to audit record data is limited to those personnel with the appropriate privileges and those authorized by appropriate ACF2 access rules. These audit records are written to the System Management Facility (SMF) datasets. ACF2 allows for the use of eight types of SMF records. These records journal the following information: invalid password use, dataset name or program violations, Logonid modifications, access rule modifications, restricted Logonid job log, TSO command records, information storage modifications, and resource violations.

During the testing, selective auditing was implemented for specified authorized accesses and for specified users. A series of authorized and unauthorized accesses were attempted, manually logged, and found to match the audit reports generated by the ACF2 report utilities. Attempts were also made to gain unauthorized access to the audit files. No unauthorized accesses were permitted.

The Utilities and Audit Manuals provide documentation on the structure and format of each audit report and guidelines on the use of these reports.

2.5 SYSTEM ARCHITECTURE

Requirement:

The Trusted Computing Base (TCB) shall maintain a domain for its own execution that protects it from external interference or tampering (e.g., by modification of its code or data structures). Resources controlled by the TCB may be a defined subset of the subjects and objects in the ADP system. The TCB shall isolate the resources to be protected so that they are subject to the access control and auditing requirements.

Satisfied By:

In the ACF2/MVS system there are several security-relevant protection mechanisms that comprise the TCB. These mechanisms are supplied by both the MVS operating system and the ACF2 add-on package.

The ACF2 system interacts with the MVS operating system through intercepts that are link-edited as front-ends to the appropriate security-relevant modules of MVS. No modifications of IBM code are required to support the standard ACF2 system.

The ACF2 security package provides by default, discretionary protection for all protected resources in the system through the use of access control lists. These access control lists are associated with every protected resource and contain the name of the resource, the user ID's of all users who are authorized to access that resource and the access type that defines in what manner those users may access the resource.

Authorization checking is done by ACF2 at all LOGONs, all attempts to READ, WRITE, ALLOCATE, SCRATCH and RENAME, and all catalog processing. Access privileges can be granted in varying degrees (READ, WRITE, ALLOCATE, and EXECUTE). These access privileges can only be assigned by the owner of the resource or a security administrator with authority over that resource.

If ACF2 is accidentally or deliberately removed from the system, a message is generated at the operator's console telling the operator that ACF2 is no longer active. When ACF2 is not active, every request to system resources appears at the operator's console and must be acted upon by the operator. The operator has the options to either allow or deny the request or to postpone it until ACF2 becomes active.

In addition to ACF2, the following are MVS/SP protection mechanisms. MVS implements two techniques to preserve the integrity of each user's work. The first is a private address space for each user and the second is the use of multiple storage protect keys.

In MVS, a virtual storage address space consists of a system area, a common area, and a private area. MVS assigns a separate address space to each user to prevent users from violating each others' address space. MVS uses multiple storage protect keys to protect the system and subsystems from unauthorized users. Before MVS performs services on the behalf of a user, it takes steps to prevent possible security violations (e.g., the use of invalid control blocks or the execution of unauthorized code) and to avoid user-induced system failures due to improperly specified requests. (NOTE: ACF2 has the ability to suppress dumps that are caused by a user process, when that process is running in a secured state (i.e., when that process is running in execute-only or restricted path mode)).

Under MVS, the information in real storage is protected from unauthorized use by means of multiple storage protect keys. A non-addressable protect key consists of a control field in storage and is associated with each 4K block of real storage. The key in storage contains the protect key of the owner and a fetch protect bit (as well as the reference and change bits maintained by the hardware and used by the software to make paging decisions).

The protect key protects the associated block of storage from unauthorized modification, while the fetch protect bit protects the block from an unauthorized attempt to read or fetch its contents. When a request is made to modify the contents of a real storage location, the key in storage is compared to the storage protection key associated with the request. If the keys match, the request is satisfied, if not, the system rejects the request and issues a program exception interrupt. When a request is made to access (read or fetch) the contents of a real storage location, the request is satisfied unless the block of storage is fetch protected. If the real storage location is fetch protected, the key in storage is compared to the key associated with the request and the resulting action is dependent upon whether or not the keys match.

Other security-relevant mechanisms that are supplied as a part of MVS are:

1. MVS provides the ability to password protect data sets. Therefore, if a user desires an extra degree of protection, (in addition to that of the ACF2 package), he can password protect his data. Different passwords can be used to allow different types of access (i.e., one password for read access, a separate password for write access, etc.).

2. The Authorized Program Facility (APF) provides the ability to limit the use of sensitive system services and resources to authorized users. The APF checks to see if the process that is requesting the system utility resides in an authorized library. If it does, the request is allowed, if not, the request is denied.

3. Another security-relevant mechanism is the provision for the erasing of main memory before it is released to any user or user's process. This prevents the scavenging of residue from main storage.

4. The Job Entry Subsystem (JES2 or 3)

   The Job Entry Subsystem has, as one of its functions, the responsibility to recover from a process error without lowering the integrity of the data that was in use at the time.

5. Utilities needed by the system

   The utility programs provided by MVS are designed to assist in organizing and maintaining data within the system. There are three classes of utility programs. They are system utilities, data set utilities and independent utilities. System utilities are used for maintaining and manipulating system and user data sets. System utilities must reside in authorized libraries and are controlled by JCL and utility control statements. Also, they can only be executed by authorized programs.

   Data set utilities are intended to be used for changing and/or comparing data at the data set or record level. They are controlled by JCL and utility control statements but can be called by any program.

   Independent utilities are used to prepare devices for system use when the operating system is not available. They operate outside of the operating system, are controlled by utility control statements and can not be called by a program (they must be run independently).

A list of the utilities supplied as a part of the MVS operating system and their functions can be found in appendix B.

Although penetration testing is not required in division C of the Criteria, various attempts were made to bypass ACF2 security mechanisms and gain unauthorized access to the ACF2 data sets. All attempts were unsuccessful.

2.6 SYSTEM INTEGRITY

Requirement:

Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the Trusted Computing Base.

Satisfied By:

An IBM supplied program can be used to verify the correct operation of the system hardware. The On-Line Test Executive Program (OLTEP) checks for the correct operation of the hardware.

In addition, MVS maintains the SYS1.LOGREC data set for the purpose of error recording. This data set is non-sharable and provides a record of all detected hardware failures and selected software errors and system conditions. Information about each incident is written into SYS1.LOGREC by the system recording routines and can be retrieved by using the environmental recording, editing and printing service aid (IFCEREP1). The IFCEREP1 output can be used for diagnostic and/or measurement purposes to maintain the devices and to support the system control program.

The IFCDIP00 service aid initializes SYS1.LOGREC during system initialization. IFCDIP00 creates a header record and a time stamp record for the SYS1.LOGREC data set and allocates space for the data set which must reside on the system residence volume.

A record is made on SYS1.LOGREC for every detected hardware or software failure and system condition that has an associated recording request or recording routine. The records contain different types of data that document failures and system conditions. The records are stored in chronological order on SYS1.LOGREC. In general, each record contains:

- Relevant system information at the time of the failure.

- Device hardware status at the time of the failure.

- Results of any device/control unit recovery attempt.

- Results of any software system recovery attempt.

- Statistical data.

There are various types of records, containing device- or incident-dependent information that can be recorded on SYS1.LOGREC, which contain complete and specific information for the device, and type of failure or system condition that caused it to be written.

Recording Machine Check records are recorded on SYS1.LOGREC whenever the following detected machine failures occur:

- Central Processing Unit (CPU) processor

- Storage

- Storage Key

- Timer

When a machine failure occurs, the Machine Check Handler (MCH) receives control via a machine-check interrupt for a soft failure (one that was corrected by the hardware retry features) or for a hard failure (one that could not be corrected by the retry features).

If the machine check interrupt is for a soft failure, MCH uses the environmental and model independent information describing the failure to build an MCH record. After formatting the information, MCH passes control to the Recovery Termination Manager (RTM). RTM then invokes the recording request routine which queues the MCH record on the asynchronous output queue and posts the asynchronous recording task. The recording task asynchronously scans the output queue and issues an appropriate SVC to write any records on this queue to SYS1.LOGREC.

If the machine check interrupt is for a hard failure, MCH analyzes the information in the model independent logout area, isolates the error, and provides a record of the analysis to RTM. RTM then takes the same actions as it does for a soft failure.

With each Initial Program Load (IPL) the system begins a sequential count of errors. The sequence number is therefore unique for each detected software error or machine failure. The sequence number remains constant for subsequent software records associated with the same error (although the time stamp may change). Software records are recorded on SYS1.LOGREC for hardware detected hardware errors, hardware detected software errors, operator detected errors and software detected software

errors. For error recording purposes, error data is collected in the System Diagnostic Work Area (SDWA) to assist in identifying the System Control Program (SCP) error and then invoke the RTM.

2.7 SECURITY TESTING

Requirement:

The security mechanisms of the ADP system shall be tested and found to work as claimed in the system documentation. Testing shall be done to assure that there are no obvious ways for an unauthorized user to bypass or otherwise defeat the security protection mechanisms of the Trusted Computing Base. Testing shall also include a search for obvious flaws that would allow violation of resource isolation, or that would permit unauthorized access to the audit or authentication data.

Satisfied By:

Testing results appear throughout this report. The security features of the ACF2/MVS system were tested and found to work as claimed in the documentation. No obvious ways to bypass the security mechanisms of ACF2 were discovered. Additional attempts were made to bypass store or fetch protection, password checking, and other ACF2 protection features in order to obtain unauthorized access or control. None of these attempts succeeded.

Ground Rule: These attempts did not involve or assume the collusion of a system programmer planting malicious code (e.g., Trojan Horse) or the assistance of a computer operator. Attempts were made to exploit either the existing code (as delivered by SKK) or code entered via a user/terminal interface.

2.8 DOCUMENTATION

2.8.1 SECURITY FEATURES USER'S GUIDE

Requirement:

A single summary, chapter or manual in user documentation shall describe the protection mechanisms provided by the Trusted Computing Base, guidelines on their use, and how they interact with one another.

Applicable Document(s):

1. A description of the protection mechanisms provided by ACF2 is contained in the ACF2 OVERVIEW (pp. 1-8) and the ACF2 General Information Manual (pp. 5-55).

2. Guidelines on the use of these mechanisms is provided in the ACF2 General Information Manual (pp. 5-55) and the ACF2 User's Guide (pp. 6-13, 16-23, 39-48).

3. The description of the interaction between the mechanisms is given in the ACF2 Installation and Maintenance Guide.

2.8.2 TRUSTED FACILITY MANUAL

Requirement:

A manual addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility. The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit event shall be given.

Applicable Document(s):

Cautions about functions and privileges that should be controlled when running a secure facility are contained in the ACF2 Implementation Planning Guide (pp. 3-15) and the Auditor's Guide (pp. 5-14 and 37-39).

2.8.3 TEST DOCUMENTATION

Requirement:

The system developer shall provide to the evaluators a document that describes the test plan and results of the security mechanisms' functional testing.

Applicable Document(s):

This requirement is satisfied by documentation that the evaluation team reviewed at SKK's offices in Rosemont, IL.

2.8.4 DESIGN DOCUMENTATION

Requirement:

Documentation shall be available that provides a description of the manufacturer's philosophy of protection and an explanation of how this philosophy is translated into the Trusted Computing Base. If the TCB is composed of distinct modules, the interfaces between these modules shall be described.

Applicable Document(s):

1. A description of the manufacturer's philosophy of protection is included in the ACF2 General Information Manual and the ACF2 Overview (pp. 1-4).

2. The description of how this philosophy is translated into the TCB is contained throughout the General Information Manual.

SECTION 3

ACF2's DEFICIENCIES AT CLASS B1

THE FOLLOWING IS A BRIEF DESCRIPTION OF HOW ACF2 COMPARES AGAINST THE CLASS B1 REQUIREMENTS OF THE CRITERIA. THIS SECTION EXPLAINS WHY ACF2 COULD NOT MEET THE REQUIREMENTS AT THE B1 LEVEL.

The ACF2/MVS system does not label subjects and objects with clearance and/or classification labels and does not possess any mandatory access control mechanism as defined in the Criteria. Therefore, the system can not satisfy the labeling and mandatory access control requirements for division B systems.

These mechanisms are the basic, fundamental features required of class B1 systems. Without these features, the system also can not satisfy the B1 Identification and Authentication requirements as the ability to validate a user's clearance level is lacking. Also, the system can not audit accesses to objects based upon the objects' security level so the system can not pass the B1 auditing requirements.

Further, the ACF2/MVS system does not satisfy the Design Specification and Verification requirements as well as the strengthened security testing and documentation requirements of class B1.

SECTION 4

ACF2 vs. REQUIREMENTS ABOVE CLASS C2

THIS SECTION ADDRESSES ONLY THOSE REQUIREMENTS THAT ACF2 SATISFIES ABOVE CLASS C2.

4.1 DISCRETIONARY ACCESS CONTROL (B3)[1]

Requirement:

The TCB shall define and control access between named users and named objects (e.g., files and programs) in the ADP system. The enforcement mechanism (e.g., access control lists) shall allow users to specify and control sharing of those objects. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of specifying, for each named object, a list of named individuals and a list of groups of named individuals with their respective modes of access to that object. Furthermore, for each such named object, it shall be possible to specify a list of named individuals and a list of groups of named individuals for which no access to the object is to be given. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users.

Satisfied By:

All resources in the system are protected by default. ACF2 provides controlled access between named users and objects by means of access rules and generalized resource rules. All accesses to the system, to data, and to resources are prevented unless specified in an access rule. Users may control the sharing of their own data between individuals as well as groups. This control can be specified (to either allow or deny access) down to the granularity of a single user by the owner of the data or a user with the SECURITY attribute.

ACF2 also provides the capability to limit access down to a specific program path (i.e., access will only be granted to a particular user if he is running a particular program from a particular library).

---

1. Although ACF2 satisfies this feature at the B3 level, ACF2 does not satisfy any of the assurance requirements above the class C2 level.

SECTION 5

EVALUATORS' COMMENTS

The audit capabilities of ACF2/MVS are versatile and extensive. With the exception of the monitoring of covert channels and auditing based on object security level, ACF2/MVS satisfies the other audit requirements of the Criteria.

ACF2/MVS also provides extensive back-up and recovery facilities for the ACF2 data sets. These facilities, however, do not satisfy the Trusted Recovery requirement of the Criteria which appears at class B3. The requirement is that a secure initial state must be attainable and it must be shown that it is possible to recover to this state, without a protection compromise, after a system failure.

For the maximum security possible, users should be defined to ACF2 with the minimum amount of authority that will allow them to efficiently and effectively use the system. Passwords should be enforced and regular password change should be required. The site should install and use the code supplied by SKK that allows for the erasure of all storage areas before these areas are released to system users. Finally, the ACF2 audit utilities should be used to the maximum, reasonable extent and the report utilities should be run regularly to obtain information about which users are utilizing what resources and how.

SECTION 6

CONCLUSIONS

The DoDCSC evaluation team has determined that the highest class at which ACF2/MVS satisfies all the requirements of the Criteria is class C2 and therefore ACF2/MVS has been assigned a class C2 rating. It should be noted, however, that this rating is contingent upon the utilization of code described in SKK's NOTE #7.

The ACF2/MVS system does not label subjects and objects with clearance and/or classification labels and does not possess any mandatory access control mechanism as defined in the Criteria. Therefore, the system does not satisfy the labeling and mandatory access control requirements for division B systems.

These mechanisms are the basic, fundamental features required of class B1 systems. Without these features, the system also does not satisfy the B1 Identification and Authentication requirements as the ability to validate a user's clearance level is lacking. Also, the system can not audit accesses to objects based upon the objects' security level so the system does not satisfy the B1 auditing requirements.

Finally, ACF2's strong discretionary access controls and audit features provide significant improvements to the security of the MVS operating system.

REFERENCES

1. Department of Defense Trusted Computer System Evaluation Criteria, Ft. Meade, MD: DoD Computer Security Center, 15 August 1983.

2. Gwatking, J.C., Automatic Erasure of Released Disk Space on an IBM 370 Computer Using the MVS Operating System, NTIS # AD-A091957, Department of Defence, Defence Research Centre, Salisbury, South Australia, June 1980.

3. Final Evaluation Report, Resource Access Control Facility (RACF) Version 1 Release 5, Ft. Meade, MD: DoD Computer Security Center, February 1984.

4. ACF2 Auditor's Guide, Doc. Nr. ABP006-01.

5. ACF2 CICS Support Manual, Doc. Nr. ABP0010-02.

6. ACF2 Command Reference Card, Doc. Nr. ABP0031-00.

7. ACF2 Composite Index, Doc. Nr. AMP0030-01.

8. ACF2 Field Definition Record Generation Manual, Doc. Nr. ABP0003-01.

9. ACF2 General Information Manual, Doc. Nr. AMG002-01.

10. ACF2 Implementation Planning Guide, Doc. Nr. ABP0012-01.

11. ACF2 IMS Support Manual, Doc. Nr. ABP0009-01.

12. ACF2 Installation and Maintenance Guide, Doc. Nr. AMP0013-02.

13. ACF2 Messages Manual, Doc. Nr. ABP0029-00.

14. ACF2 Other Products Manual, Doc. Nr. ABP0011-00.

15. ACF2 Overview, Doc. Nr. ABG0001-01.

16. ACF2 System Programmer's Guide, Doc. Nr. AML0007-01.

17. ACF2 User Modifications Catalog, Doc. Nr. ACG3005-01.

18. ACF2 User's Guide, Doc. Nr. ABP0005-01.

19. ACF2 Utilities Manual, Doc. Nr. ABP0004-01.

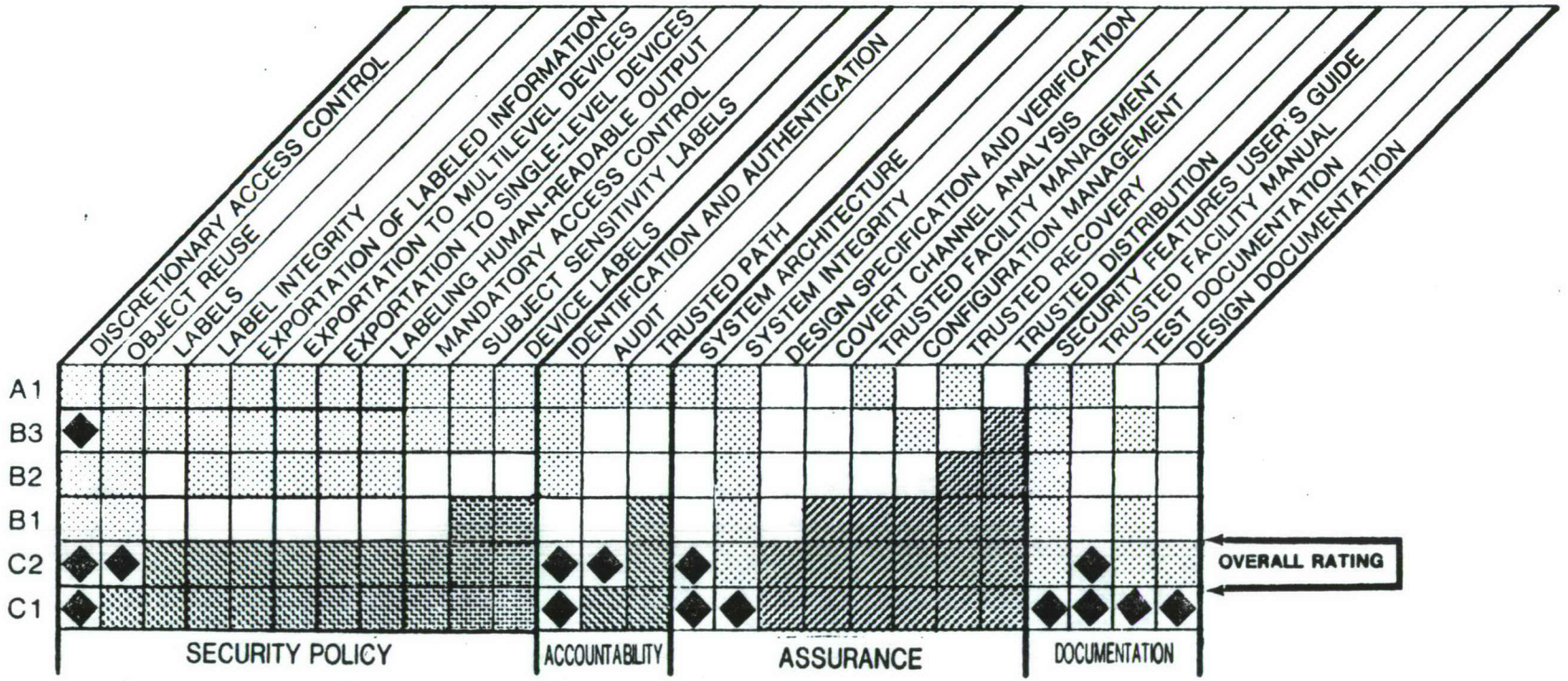
20. OS/VS2 MVS Supervisor Services and Macro Instructions, GC28-0683-2.

GLOSSARY

The following is a list of acronyms used in this report along with the page number on which they are first used.

ADP	Automatic Data Processing, 5
CICS	Customer Information Control System, 6
CPU	Central Processing Unit, 13
DoDCSC	Department of Defense Computer Security Center, iv
EPL	Evaluated Products List, 1
FDR	Field Definition Record, a-1
FY	Fiscal Year, 1
IMS	Information Management System, 6
IPL	Initial Program Load, 13
JCL	Job Control Language, 7
JES	Job Entry Subsystem, 2
MCH	Machine Check Handler, 13
MVS	Multiple Virtual Storage, iv
MVS/SP	Multiple Virtual Storage/System Product, iv
OLTEP	On-Line Test Executive Program, 12
RTM	Recovery Termination Manager, 13
SCP	System Control Program, 13
SDWA	System Diagnostic Work Area, 13
SMF	System Management Facility, 3
SPF	System Productivity Facility, 2
TCB	Trusted Computing Base, 5
TSO	Time Sharing Option, 2

TRUSTED COMPUTER SYSTEM EVALUATION SUMMARY CHART



NO REQUIREMENTS FOR THIS CLASS

NO ADDITIONAL REQUIREMENTS FOR THIS CLASS

MEETS OR EXCEEDS THE REQUIREMENTS FOR THIS CLASS

SYSTEM NAME: the Access Control Facility 2 (ACF2)
VENDOR: SKK, Inc.
EVALUATION DATE: 3 August 1984

APPENDIX A

1.0 TEST PLAN FOR ACF2 vs. CLASS C2 REQUIREMENTS

BEFORE ANY TESTS ARE STARTED, RUN ALL OPTIONS OF THE SHOW COMMAND TO DETERMINE THE 'STATUS' OF THE SYSTEM. OBTAIN THE DEFAULT FDR AS IT IS DEFINED BY THE INSTALLATION.

```
/ ACF
/ SHOW ACTIVE/DDSN/FIELDS/MODE/PROGRAMS/RESIDENT/STATE/
       SYSTEMS/TSO/ZEROFLDS
```

1.1 DISCRETIONARY ACCESS CONTROL

Detailed Test Plan:

Develop an access control matrix defining users vs. resources with authorized access rights, user groups, and resource groups, and implement via proper ACF2 access rules.

user	user attributes	user privileges/limitations (THAT ARE NOT DEFAULT)
user1	SECURITY	NON-CNCL JCL MAIL TSO
user2	ACCOUNT	JCL TSO
user3	LEADER	JCL TSO
user4	AUDIT	TSO
user5		MONITOR NO-STORE RULEVLD TSO-TRC
user6		JCL TSO TRACE
user7	SECURITY	INTERCOM JCL TSO
user8		RULEVLD TRACE TSO-TRC
user9		EXPIRE(enter date) JCL TSO
user10	AUDIT	NON-CNCL JCL TSO
user11	ACCOUNT	JCL TSO
user12		MAXDAYS(2)
user13		MAXDAYS(01) MINDAYS(10) NO-STORE
user14	ACCOUNT	
user15		TSO TSO-TRC
user16	SECURITY	READALL JCL TSO
user17		MINDAYS(0)
user18	AUDIT	JCL TSO
user19		READALL RULEVLD JCL TSO
user20	LEADER	JCL TSO
user21		PMT-ACCT TSO-TRC
user22		NON-CNCL
user23		
user24		SHIFT(tstshift) **create shift record
user25		SOURCE(terminal #)
user26		TRACE TSO
user27		TSO TSO-TRC

```
user28                          TSO TSOTIME(10)
user29     CONSULT              NON-CNCL
user30                          OPERATOR
```

*** USE A DEFAULT USERID FOR ALL UNPRIVILEGED USERS ***

*** enter the following commands under TSO to create the users defined in the matrix

```
/ acf
/ set lid

/ insert using(default logonid) new-userid  *privileges*
        THE DEFAULT LOGONID SHOULD BE ONE USED TO DEFINE
        UNPRIVILEGED USERS

/ insert user1  PASSWORD(user1he7) SECURITY NON-CNCL JCL
                MAIL TSO
/ insert user2  PASSWORD(user2fd1) ACCOUNT JCL TSO
/ insert user3  PASSWORD(user3bur) LEADER JCL TSO
/ insert user4  PASSWORD(user4ndi) AUDIT TSO
/ insert user5  PASSWORD(user5hu3) MONITOR NO-STORE RULEVLD
                TSO-TRC
/ insert user6  PASSWORD(user6dp9) JCL TSO TRACE
/ insert user7  PASSWORD(user7gtr) SECURITY INTERCOM JCL TSO
/ insert user8  PASSWORD(user8mww) RULEVLD TRACE TSO-TRC
/ insert user9  PASSWORD(user9dko) EXPIRE(      ) JCL TSO
/ insert user10 PASSWORD(user10fe) NON-CNCL AUDIT JCL TSO
/ insert user11 PASSWORD(user11ng) ACCOUNT JCL TSO
/ insert user12 PASSWORD(user12fr) MAXDAYS(1)
/ insert user13 PASSWORD(user13vy) MAXDAYS(01) MINDAYS(10)
                NO-STORE
/ insert user14 PASSWORD(user14di) ACCOUNT
/ insert user15 PASSWORD(user15ye) TSO TSO-TRC
/ insert user16 PASSWORD(user16f4) SECURITY READALL JCL TSO
/ insert user17 PASSWORD(user17dk) MINDAYS(0)
/ insert user18 PASSWORD(user18ht) AUDIT JCL TSO
/ insert user19 PASSWORD(user19be) READALL RULEVLD JCL TSO
/ insert user20 PASSWORD(user2010) LEADER JCL TSO
/ insert user21 PASSWORD(user21ms) PMT-ACCT TSO-TRC
/ insert user22 PASSWORD(user22nt) NON-CNCL
/ insert user23 PASSWORD(ulogony2)
/ insert user24 PASSWORD(user24sc) SHIFT(tstshift)
/ insert user25 PASSWORD(user25fd) SOURCE(      )
/ insert user26 PASSWORD(user26f6) TRACE TSO
/ insert user27 PASSWORD(user27gR) TSO TSO-TRC
/ insert user28 PASSWORD(user28HR) TSO TSOTIME(10)
/ insert user29 PASSWORD(user29G3) CONSULT NON-CNCL
/ insert user30 PASSWORD(user302K) OPERATOR
```

Insure that newly created objects are protected by default under ACF2.

```
*** using various userids, create several data sets and
    write some access rules for these data sets.  Verify
    that no-one is allowed access unless they are included
    in an access rule.


/ $KEY(***) data.set.name UID(***) PROGRAM(???) FOR(1) R(L)
                          W(L)
            data.set.name UID(***) R(A) W(A)


    *** WRITE A PROGRAM USING user6 THAT WILL ATTEMPT TO
        ACCESS ONE OF THESE DATA SETS, USE THIS PROGRAM TO
        TEST THE EXECUTE ONLY AND PROGRAM PATHING FEATURES

   *****ON THE TEST SYSTEM, THE UID=THE LOGONID*****

/   $KEY(user1)
    %CHANGE *UID*
    test1.data UID( ) R(P) W(P) A(P)
    test1.data UID( ) R(A) W(A) A(P)
    test1.data UID( ) R(L) W(P) A(L)

/   $KEY(user2)
    test2.data UID( ) R(L) W(L) A(P)
    test2.data UID( ) R(P) W(P) A(P)
    test2.data UID( ) R(P) W(A) A(P)
    - R(A) W(P) A(P) E(P)

/   $KEY(user3)
    test3.data UID( ) R(P) W(L) A(P)
    test3.data UID( ) R(A) W(P) A(L)
    test3.data UID( ) R(A) W(A) A(A)
    - R(P)

/   $KEY(user4)
    test4.data UID( ) R(P) W(P) A(P)
    test4.data UID( ) R(P) W(P) A(P)
    test4.data UID(*) R(A)
    test4.data UID( ) LIB( ) PGM( ) R( ) W( ) A( )

/   $KEY(user5)
    test5.data UID( ) R(L) W(L) A(L)
    test5.data UID( ) R(P) W(P) A(P)
    test5.data UID( ) R(A) W(L) A(P)

/   $KEY(user6)
    test6.data UID( ) R(P) W(P) A(P)
    test6.data UID( ) R(P) W(P) A(P)
    test6.data UID( ) R(A) W(L) A(P)

*** WRITE AN ACCESS RULE TO TEST THE PROGRAM PATHING FEATURE***
```

```
/   $KEY(user1)
    test1.data UID( ) PROGRAM(testprog) R(A) W(L)
```

Write an access rule for system resources and then attempt to bypass ACF2 protection of system resources.

```
/   $KEY(SYS1)
    UADS UID( ) R(A) W(L) A(L)
    UADS UID(*) R(P) W(P) A(P)
    PARMLIB UID( ) R(A) W(L) A(L)
    PARMLIB UID( ) R(A)
    PARMLIB
    MAN* UID( ) R(A) W(L) A(P)
    MAN*
    - R(A) E(A)
```

1.2 OBJECT REUSE

Detailed Test Plan:

Using JCL, allocate disk space and attempt to print its contents. In order to verify that another user's residue can be obtained, create a very large file of recognizable information. Expand that file until it fills the entire disk space of the owner. Next delete the file, and log on as a different user in a different group who did not have access to the deleted data set. This new user shall not possess any special attributes that would enable him to bypass ACF2/MVS protection. Execute the scavenging JCL and check to see if there are pieces from the previously deleted data set.

1.3 IDENTIFICATION AND AUTHENTICATION

Detailed Test Plan:

Insure the provision of support for password expiration dates, and required password change.

*** users 12 and 17 can be used to test the maximum and minimum number of days before passwords must be changed

*** insure that the user MUST change his/her password at the end of the maxdays period. Also, insure that the user can not change his/her password before the mindays period has passed. Insure that when a user is allowed to change his/her password, it must comply with the password syntax rules.

Attempt illegal accesses to ACF2 data sets to obtain encrypted passwords, or to add, modify, or delete profile data.

Set a limit on the number of consecutive incorrect logon attempts. Exceed this limit using legal userid and incorrect password and insure that these unsuccessful attempts cause the userid to be suspended, and that messages concerning these unsuccessful logon attempts are sent to the system or security console.

*** implement and observe

*** what does ACF2 do when an incorrect logonid is used?
is the operator and/or security officer notified?
if so, when? can this be audited?

NOTE: MESSAGE ROUTING CAN BE AFFECTED BY THE OPTIONS SPECIFIED IN THE FDR

Insure that passwords are not echoed back to terminals (i.e., password suppression).

*** observe

1.4 AUDIT

Detailed Test Plan:

Implement selective auditing of specified authorized accesses and unauthorized access attempts by specified users and for specified resources. Manually log a series of authorized and unauthorized accesses. Invoke the appropriate ACF2 report utility to generate audit reports. Verify the reports by comparing to the manual log.

*** Implement selective auditing and attempt legal and illegal accesses, manually log these accesses.

```
//REPORT   EXEC PGM=ACFRPTCR,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(TSO COMMAND STATISTICS)')
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR


//REPORT   EXEC PGM=ACFRPTDS,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(DATA SET ACCESS JOURNAL)')
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR
```

```
//REPORT   EXEC PGM=ACFRPTEL,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(INFO STOR UPDATE)')
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR

//REPORT   EXEC PGM=ACFRPTIX,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(ACCESS INDEX REPORT)')
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR

//REPORT   EXEC PGM=ACFRPTJL,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(RESTRICTED LOGONID JOB LOG)')
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR

//REPORT   EXEC PGM=ACFRPTLL,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(LOGONID MODIFICATION LOG)')
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR

//REPORT   EXEC PGM=ACFRPTPW,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(INVALID PASSWORD AUTHORITY)')
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR

//REPORT   EXEC PGM=ACFRPTRL,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(RULE-ID MODIFICATION LOG)')
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR

//REPORT   EXEC PGM=ACFRPTRV,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(GENERALIZED RESOURCE LOG)')
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR
```

```
//REPORT   EXEC PGM=ACFRPTRX,REGION=640K,
//         PARM='DSET,LID(      )'
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   UNIT=3330,SPACE=(CYL(2,2)),DCB=BUFNO=30
//SYSUT2   DD   UNIT=3330,SPACE=(CYL(2,2)),DCB=BUFNO=30


//REPORT   EXEC PGM=ACFRPTSL,REGION=128K,
//         PARM=('SDATE(     ),EDATE(     )',
//         'TITLE(LOGONID SUPERLIST REPORT)',
//         INPUT(SMF),REPORT(FULL))
//SYSPRINT DD   SYSOUT=A
//RECMANX  DD   DSN=SYS1.MANX,DISP=SHR
//RECMANY  DD   DSN=SYS1.MANY,DISP=SHR


//REPORT   EXEC PGM=ACFRPTXR,REGION=640K,
//         PARM='DSET,ACF2,RRSUM,DSN(     )'
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   UNIT=3330,SPACE=(CYL(2,2)),DCB=BUFNO=30
//SYSUT2   DD   UNIT=3330,SPACE=(CYL(2,2)),DCB=BUFNO=30
```

Attempt to obtain illegal access to audit files. Attempt to read and/or write to SYS1.MAN1 and/or SYS1.MAN2. Try to delete one of these data sets. ARE THESE DATA SETS PROTECTED BY DEFAULT UNDER ACF2 OR DO THEY HAVE TO BE PROTECTED AT SYSGEN???

/ ***

Insure that audit records contain for each entry; the userid, resource(s), type of access attempted or obtained, and time of access.

*** Examine the reports and manual log generated earlier in this test plan and verify that the audit records contain the required information.

Verify that individual accountability is enforced (i.e., insure that a user's illegal or questionable actions, or actions for which a form of TRACE has been requested, are always logged as being performed by that user).

*** Examine the audit records and verify that all accesses and access attempts listed in the manual log were recorded as being made/attempted by the user indicated in the log.

*** Verify that all system accesses by users 6, 8, and 26 are logged.

*** Verify that every use of a TSO command by users 5, 8, 15, 21, and 27 are logged.

NOTE: LOGGING CAN ALSO BE SPECIFIED FOR SPECIFIC DATA SETS AND PROGRAMS. THESE FEATURES SHOULD ALSO BE TESTED.

1.5 RESOURCE ENCAPSULATION

Detailed Test Plan:

Attempt to obtain each and every type of access to each ACF2 protected resource defined in the access control matrix (implemented in section 1.1 of this test plan). Insure that only those accesses defined as legal by the matrix are allowed and that all other access attempts are prohibited and appropriately logged in the SMF records and that complete reports are available via the ACF2 report utilities.

Ground Rule: These attempts will not involve or assume the collusion of a system programmer planting malicious code (e.g., Trojan Horse) or the assistance of a computer operator. Attempts may be made to exploit either the existing code (as delivered by SKK) or code entered via a user/terminal interface.

*** This requirement is satisfied by earlier tests in this test plan, (sections 1.1 and 1.4).

ACF2 provides the ability to grant a user execute only access. This feature is not called for in the criteria but may be tested and included in the report for "comment only".

/ *** Allow execute only access to a specified program for a specified user, insure that the specified user can not obtain any access other than execute.

```
/ $KEY(***)
  progl.test UID(***) E(A) R(P) W(P)
  progl.test UID(***) E(L) R(P) W(P)
```

1.6 SYSTEM ARCHITECTURE

Detailed Test Plan:

Insure that system level data structures, both MVS and ACF2, are afforded protection against unauthorized modification.

*** The test for this requirement is included in previous sections of this test plan (sections 1.1 and 1.3).

*** Make additional attempts to modify or corrupt the SYS1. data sets.

1.7 SYSTEM INTEGRITY

Detailed Test Plan:

Insure that the IBM-supplied MVS utility programs provide sufficient assurance of correct operation.

*** Verify by inspection that IBM-supplied MVS utility programs provide sufficient assurance of correct operation.

Provide a test to check for operation of ACF2 intercepts when ACF2 main task is not active.

/ p acf2

*** attempt to access a data set and insure that the request appears at the operator's console and that the operator must act to allow, deny or postpone (until ACF2 is restarted) any access request.

*** examine how ACF2 handles started tasks as well as new jobs

1.8 SECURITY TESTING

Detailed Test Plan:

Using the documentation, determine which security features have not already been included in the test plan and develop tests to insure that these features function as claimed. (NOTE: IT WILL NOT BE POSSIBLE TO TEST EVERY DIFFERENT CONFIGURATION OF THE FDR)

Make additional attempts to bypass store or fetch protection, password checking, and other ACF2 protection features in order to obtain unauthorized access or control.

Ground Rule: These attempts will not involve or assume the collusion of a system programmer planting malicious code (e.g., Trojan Horse) or the assistance of a computer operator. Attempts may be made to exploit either the existing code (as delivered by SKK) or code entered via a user/terminal interface.

Inspect audit files to insure that the above actions have been logged according to the audit requirements specified for the encapsulated resources.

1.9 DOCUMENTATION

Detailed Test Plan:

Examine the listed ACF2 documents and determine if the class C2 documentation requirements (1.9.1-1.9.4) are satisfied.

1.9.1 SECURITY FEATURES USER'S GUIDE

- ACF2 Overview (pp. 1-8) and General Information Manual (pp. 5-55) provide descriptions of the protection mechanisms.
- General Information Manual (pp. 5-55) and User's Guide (pp. 6-13, 16-23, 39-48) provide guidelines on the use of the protection mechanisms.
- Installation and Maintenance Guide descibes the interaction between the protection mechanisms.

1.9.2 TRUSTED FACILITY MANUAL

- Implementation Planning Guide and Auditor's Guide provide cautions about functions and privileges that should be controlled when running a secure facility.
- System Programmer's Guide provides the detailed audit record structure.

1.9.3 TEST DOCUMENTATION

SKK internal use only documents

1.9.4 DESIGN DOCUMENTATION

- General Information Manual and ACF2 Overview (pp. 1-4) provide a description of the manufacturer's philosophy of protection.

2.0 TEST PLAN FOR ACF2 vs. HIGHER LEVEL REQUIREMENTS

2.1 DISCRETIONARY ACCESS CONTROL

Detailed Test Plan:

Develop an access control matrix defining users vs. resources with authorized access rights, user groups, and resource groups, and implement by writing the proper ACF2 access and resource rules.

*** THIS REQUIREMENT IS TESTED IN SECTION 1.1 OF THIS TEST PLAN.

Insure that these controls can be implemented down to the granularity of a single user.

*** Write access rules to allow or deny access to a single user and verify that these access rules are enforced.

2.2 IDENTIFICATION AND AUTHENTICATION

Detailed Test Plan:

Attempt different accesses, both authorized and unauthorized, by different users, groups, etc. Manually log these access attempts and use the ACF2 Report utilities to examine the SMF records to insure that all accesses and attempts have been properly accounted and recorded.

*** THIS REQUIREMENT IS INCLUDED IN PREVIOUS SECTIONS OF THIS TEST PLAN (SECTIONS 1.3 AND 1.4).

2.3 AUDIT

Detailed Test Plan:

Implement selective auditing of specified authorized accesses and unauthorized access attempts by specified users and for specified resources. Manually log a series of authorized and unauthorized accesses. Invoke the appropriate ACF2 report utility and generate audit reports. Check these reports for correctness by comparing to the manual log.

Attempt to obtain illegal access to audit files. Attempt to circumvent auditing for a specified access attempt by first overloading audit files by appending data to the audit files and/or by creating a series of illegal access attempts prior to the specified access.

*** This can be prevented by utilizing the MAXVIO option in the FDR

Insure that audit records contain for each entry; the userid, resource(s), type of access attempted or obtained, input source, and time and date of access.

Verify that individual accountability is enforced (i.e., insure that a user's illegal or questionable actions, or actions for which a form of TRACE has been requested, are always logged as being performed by that user).

Insure that violation attempts and logonid suspensions are immediately indicated to the operator and/or security console(s).

*** ALL OF THESE REQUIREMENTS ARE TESTED AT EARLIER STAGES OF THE TEST PLAN, (SECTIONS 1.3 AND 1.4).

APPENDIX B

The following are the utilities provided as a part of the MVS operating system:

System:

IEHATLAS -- assigns alternate tracks and recovers data when defective tracks are indicated.

IEHDASDR -- initializes and labels direct access volumes, assigns alternate tracks when defective tracks are indicated, or dumps or restores data.

IEHINITT -- writes standard labels on tape volumes

IEHLIST -- lists system control data

IEHMOVE -- moves or copies data

IEHPROGM -- builds and maintains system control data

IFHSTATR -- selects, formats, and writes tape errors from the SYS1.MAN data sets

Data Set:

IEBCOMPR -- compares records in data sets

IEBCOPY -- copies, compresses, or merges partitioned data sets

IEBDG -- creates a test data set consisting of patterned data

IEBEDIT -- selectively copies job steps and their job statements

IEBGENER -- copies records from a sequential data set

IEBISAM -- reformats source data in one sequential data set and places it in another for reconstruction

IEBPTPCH -- prints or punches records in a data set

IEBTCRIN -- constructs records from input data read from an IBM 2495 Tape Cartridge Reader

IEBUPDTE -- incorporates changes into data sets

Independent: IBCDASDI -- initializes DASD volumes and assigns alternate tracks

IBCDMPRS -- dumps and restores the contents of a DASD volume

ICAPRTBL -- loads form control and character set buffers of a 3211